Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of claims:

Claim 1 (currently amended). A method for performing a cryptographic algorithm, the cryptographic algorithm including a [[of]] modular multiplication of a multiplicand by a multiplier ~~within a cryptographic algorithm~~, in which a modulus is employed, wherein the multiplicand, the multiplier, and the modulus are parameters in the cryptographic algorithm, using a multiplication look-ahead process and a reduction look-ahead process, the method comprising the steps of:

transforming the modulus into a transformed modulus being greater than the modulus by multiplying the modulus by a transforming number, the transforming number being calculated using the modulus such that a predetermined fraction of the transformed modulus has a higher-order digit with a first predetermined value followed by at least one lower-order digit having a second predetermined value;

iteratively working off the modular multiplication using the multiplication look-ahead process and the reduction look-ahead process and utilizing the transformed modulus so as to obtain

at the end of the iteration a transformed result for the modular multiplication, the predetermined fraction of the transformed modulus being used in the reduction look-ahead process; and

re-transforming the transformed result by modular reduction of the transformed result utilizing the modulus.

Claim 2 (previously presented). The method according to claim 1, wherein the step of iteratively working off comprises a plurality of iteration steps, with a multiplication intermediate result and a reduction shift value being determined in one of the iteration steps, with the reduction shift value being computed using a determination of the number of digits between the higher-order digit with the first predetermined value of the transformed modulus and the highest-order digit of the intermediate result having ~~said~~ the first predetermined value.

Claim 3 (previously presented). The method according to claim 2, which further comprises determining a multiplication shift value in the multiplication look-ahead process, and calculating the reduction shift value for the reduction look-ahead process by subtraction of the predetermined number of digits from the multiplication shift value.

Claim 4 (previously presented). The method according to claim 1, wherein the step of iteratively working off comprises the following steps:

in a first iteration step:

(a) performing the multiplication look-ahead process to obtain a multiplication shift value;

(b) multiplying a base raised to the power of the multiplication shift value by a current intermediate result to obtain a shifted intermediate result;

(c) performing the reduction look-ahead process to obtain a reduction shift value by determining an auxiliary shift value equal to the number of digits between the higher-order digit with the first predetermined value of the predetermined fraction of the transformed modulus and the highest-order digit of the intermediate result having the first predetermined value, and by calculating the reduction shift value using the auxiliary shift value and the multiplication shift value;

(d) multiplying the transformed modulus by the base raised to the power of the reduction shift value to obtain a shifted transformed modulus; and

(e) summing the intermediate result and the multiplicand and subtracting the shifted transformed modulus to obtain an updated intermediate result.

Claim 5 (previously presented). The method according to claim 1, wherein said predetermined fraction of the modulus is 2/3.

Claim 6 (previously presented). The method according to claim 5, wherein the multiplicand, the multiplier and the modulus are binary, with the base being 2, and the higher-order digit of the predetermined fraction of the transformed modulus has the first predetermined value of 1 and the at least one low-order digit has the second predetermined value of 0.

Claim 7 (previously presented). The method according to claim 6, wherein the most significant bit of the transformed modulus is a sign bit, and a higher-order section of the predetermined fraction of the modulus reads as follows:

01000 xx ... xx,

in which the bits designated xx have arbitrary values.

Claim 8 (previously presented). The method according to claim 7, wherein the higher-order section of the transformed modulus reads as follows:

01100 ... 00.

Claim 9 (previously presented). The method according to claim 1, wherein the step of transforming the modulus comprises randomization of the modulus so that the transformed modulus is randomized.

Claim 10 (currently amended). A processor for performing a cryptographic algorithm, the cryptographic algorithm including a modular multiplication of a multiplicand by a multiplier ~~within a cryptographic algorithm~~, in which a modulus is employed, wherein the multiplicand, the multiplier, and the modulus are parameters in the cryptographic algorithm, using a multiplication look-ahead process and a reduction look-ahead process, comprising:

a ~~transformer~~ means for transforming the modulus into a transformed modulus being greater than the modulus by multiplying the modulus by a transforming number, the transforming number being calculated using the modulus such that a predetermined fraction of the transformed modulus has a higher-order digit with a first predetermined value followed by at least one lower-order digit having a second predetermined value;

a ~~processor~~ means for iteratively working off the modular multiplication using the multiplication look-ahead process and the reduction look-ahead process and utilizing the transformed modulus so as to obtain at the end of the iteration a transformed result for the modular multiplication, the predetermined fraction of the transformed modulus being used in the reduction look-ahead process; and

a ~~re-transformer~~ means for re-transforming the transformed result by modular reduction of the transformed result utilizing the modulus.

Claim 11 (currently amended). The processor according to claim 10, comprising a host CPU and a coprocessor, said means for transforming the modulus ~~transformer~~ being arranged in the host CPU and said means ~~processor~~ for iteratively

working off the modular multiplication being arranged in the coprocessor.

Claim 12 (previously presented). The processor according to claim 11, wherein the host CPU is a short-number arithmetic-logic unit having a number of digits smaller than or equal to 64, and the coprocessor is a long-number arithmetic-logic unit having a number of digits greater than or equal to 512.

Claim 13 (currently amended). The processor according to claim 10, wherein the means ~~processor~~ for iteratively working off the modular multiplication includes a register for the transformed modulus and a register for an intermediate result of the modular multiplication.